Consultation Paper on the Selection of a Block Cipher Based MAC Algorithm

The latest step in the efforts of the National Institute of Standards and Technology (NIST) to develop and update block cipher modes of operation was the proposal of the RMAC algorithm in the draft NIST Special Publication 800-38B. NIST received several public comments on the draft that raised security issues, and [Rog] and [Wag] explicitly called for NIST to withdraw RMAC in favor of some other algorithm. This note summarizes the technical issues underlying the selection of RMAC and proposes modifications to the draft for further public comment.

The main proposed modification is to remove the salt *R* from two general-purpose parameter sets, so that in those cases the algorithm is equivalent to the EMAC algorithm, which is described and analyzed in [PR]. The two remaining parameter sets that include *R* would be retained, with an option for the RMAC submitters' "mode 2" construction, which was omitted from the initial draft.

This note assumes that the reader is familiar with the MAC algorithms in question and with the FIPS-approved block cipher algorithms: AES and Triple DES (TDES). While AES is the preferred block cipher algorithm for its performance and security attributes, TDES with two or three keys is still approved and used in many applications. Therefore, NIST is considering TDES in the selection of a block cipher based authentication mode, recognizing that the submitted proposals were not specifically designed for TDES.

The submitted modes proposals and the public comments that NIST received are available through http://csrc.nist.gov/encryption/modes/.

The Initial Selection of RMAC

NIST has received several proposals for MAC algorithms based on block ciphers. To a first approximation, the proposals require a similar number of invocations of the underlying block cipher. The performance of two proposals, the XECB-MAC family and PMAC, stands out, especially for high-end applications, because their block cipher invocations are parallelizable; however, patents are pending on the techniques in these proposals. Because there is strong and widespread opposition to patent-encumbered techniques, as expressed, for example, at the two public workshops on modes of operation, NIST has decided not to recommend XECB-MAC or PMAC at this time.

The two other MAC proposals are RMAC and XCBC; since the second workshop, NIST also has received proposals for TMAC and OMAC, variations of XCBC with two keys and one key, respectively. The RMAC algorithm is a randomized variation of the EMAC algorithm. EMAC was not directly submitted to NIST, but it is an established algorithm, essentially equivalent to MAC Algorithm 2 in ISO 9797-1.

NIST selected RMAC over XCBC because, although XCBC offers somewhat better performance, NIST believed that RMAC could be used securely for more messages than could XCBC, and because XCBC seemed less suited for use with TDES. Subsequently, public comments have called into question the level of security assurance that the NIST draft of RMAC offers both for AES and, especially, TDES. Revising the specification to focus on the underlying EMAC construction of RMAC appears to address the most significant concerns.

The security, performance, and keys of RMAC, EMAC, and XCBC are discussed below. TMAC and OMAC are discussed in connection with key storage and derivation; otherwise, they are similar to XCBC.

Security

The following is an informal summary of the security of RMAC, EMAC, and XCBC:

- *Improved Variants of CBC-MAC*: RMAC, EMAC, and XCBC are designed to prevent the straightforward "concatenation" forgeries to which CBC-MAC (specified, for example, as MAC Algorithm 1 in ISO 9797-1) is vulnerable when used for variable length messages.

- *Proof Models*: The RMAC and XCBC proposals provide proofs of security properties, but under different mathematical models. In [Rog] and [Bl] (and also supported in [Wag]), the XCBC submitters assert that the standard assurance, "reduction-based provable security," such as provided by XCBC, is clearly preferable, when available, to the "ideal cipher model" of assurance provided by RMAC.

  The RMAC submitters in [JJV2] assert that AES is likely to meet the stronger assumptions implied by the ideal cipher model. They also point out that reduction-based assurance is provided by their optional "mode 2" construction that was omitted from the NIST draft.

  EMAC, like XCBC, offers reduction-based security assurance. At NIST's request, the RMAC submitters have documented in [JJV1] that RMAC inherits this type of assurance from EMAC, with bounds comparable to EMAC's bounds, as described in the next bullet.

- *Proof Bounds with AES*: According to the summary in the RMAC submission, the advantage of an attacker in forging a (new) MAC is bounded by $(517L+t)/2^{128}$, where L is the total length of queries to RMAC generation and verification oracles, involving t invocations of AES by the oracles and the attacker. Consequently, for any foreseeable values of L and t, the attacker's advantage in the model of the proof would be negligible.

  The analogous bound for EMAC is $2L^2/2^{128}$. Consequently, for almost any practical value of L, the attacker's advantage in the model of the proof would be extremely small.

  According to the summary in the XCBC submission, with AES, given q XCBC oracle queries of length at most m blocks, the attacker's probability of forging a (new) MAC is bounded by $5m^2q^2/2^{128}$. Consequently, the XCBC proof does not appear to provide assurance against forgery if, for example, an attacker can collect the MACs of 2^{32} messages, at least one of which consists of 2^{31} blocks. However, given a moderate limit on m, the attacker's advantage in the model of the proof would be extremely small for almost any practical value of q.

- *Proof Bounds with TDES*: Because the block size of TDES is 64 bits, it is an inherently problematic building block for a modern MAC algorithm. NIST originally believed that RMAC addressed this deficiency. However, the RMAC submitters caution in [JJV2] that using RMAC with TDES greatly damages the security of the RMAC construction. A similar concern is implied in [Ll] and illustrated by a practical attack in [Kn] on RMAC with TDES that exploits an ambiguity in the NIST draft. The ambiguity can be resolved in a manner that precludes this particular attack, although it is an open question whether sufficient security assurance can be provided in this case.

  The security bound for XCBC with TDES, analogous to that in the previous bullet, is $5m^2q^2/2^{64}$, which probably does not provide adequate assurance for general applications.

  The security bound for EMAC with TDES, analogous to that in the previous bullet, is $2L^2/2^{64}$, which provides some assurance for general applications.

- *Design Principles*: Several commenters expressed concerns about the wisdom and maturity of the RMAC design (especially the varying of the second block cipher key) in [Ll], [Kn], [Rog], [Koh], [JJV2], and [Bl], including impractical attacks in [Ll], [Kn], and [Koh].

- *Changes to RMAC in the NIST Draft*: The XCBC submitters in [Rog] and [Bl] also assert that changes to RMAC in the NIST draft may invalidate its security assurance. NIST assessed and

coordinated some of these changes—the guidelines for parameter settings, the key derivation option, and the option for *R* to be a nonce, but not the use of TDES—with the RMAC submitters, who have further documented the security assurance in [JJV1].

Performance

In performance, XCBC is preferable to RMAC, and, to a lesser extent, to EMAC:

- *Key Setup*: XCBC requires the setup of the block cipher under a single, fixed key. RMAC and EMAC require the setup of the block cipher under two keys. For EMAC, both keys are fixed; for RMAC, the first key is fixed, and the second key may vary across messages

- *Block Cipher Invocations*: RMAC and EMAC require, per message, one or two more invocations of the block cipher than XCBC to compute the MAC. This overhead is significant for applications involving relatively short data.

- *Tag Size*: For a given MAC length, RMAC authentication tags are longer, when *R* is used, than those of XCBC and EMAC.

- *Overhead for R*: In the submitted RMAC proposal, RMAC requires a random number generator for *R*; XCBC and EMAC have no such requirement.

The draft SP 800-38B of RMAC allows *R* to be a nonce; thus, instead of random number generation, the overhead for *R* could take the form of maintaining state to keep track of the nonce.

Key Storage and Key Derivation

RMAC and EMAC each require two block cipher keys, and XCBC requires one block cipher key and two block-length keys; thus, a comparison of the key storage requirements depends on the block size and the key size of the block cipher. The possibility of incorporating key derivation methods into the modes further complicates the comparison. In the case of RMAC, NIST suggested in the draft SP 800-38B a method for deriving the two RMAC keys from a single master key, with computational overhead of 2-6 invocations of the block cipher, depending on its block size and key size. In the case of XCBC, the TMAC proposal derives the second block-length key from the first block-length key, with negligible overhead (i.e., much less than a block cipher invocation), and the OMAC proposal derives both block-length keys using the block cipher key, with an overhead of, essentially, one block cipher invocation.

Table 1 below identifies, for each approved block cipher algorithm, the number of key bits that are required in variations of RMAC and XCBC. For TMAC, OMAC, and the RMAC key derivation option discussed above, the minimum key bits for storage/transmission are listed, i.e., prior to the derivation of the XCBC/RMAC keys. The submitters' "mode 2" construction, omitted from the NIST draft, which specifies AES256 for the final invocation of the block cipher, is also included, with and without the key derivation option.

Table 1: Key Requirements (bits)

	AES128	AES192	AES256	TDES112	TDES168
RMAC	256	384	512	224	336
RMAC + key derivation	128	192	256	112	168
RMAC Mode 2	384	N/a	N/a	N/a	N/a
RMAC Mode 2 + key der.	128	N/a	N/a	N/a	N/a
XCBC	384	448	512	240	296
TMAC	256	320	384	176	232
OMAC	128	192	256	112	168

Moving Forward

NIST further evaluated its selection of a block cipher authentication mode for AES and for TDES in light of the public comments on the draft SP 300-38B.

For TDES, EMAC appears to offer greater security assurance than RMAC and XCBC, which, in fairness, were not specifically designed for use with TDES. In particular, the attacks and other important security concerns expressed in the public comments about RMAC do not seem to apply to EMAC, and EMAC's security bounds are much better than those of XCBC, under the same, standard model of proof. Although these bounds may still not offer sufficient assurance for many modern applications, this deficiency is essentially a consequence of the 64 bit block size of TDES, which would be best addressed by migration to AES.

For AES, the choice of RMAC or EMAC over XCBC amounts to a tradeoff of performance for security assurance. XCBC offers better performance in several respects than RMAC and, to a lesser extent, than EMAC, especially for short messages. RMAC inherits from EMAC security bounds that are better than those of XCBC, and under the same, standard model of proof; moreover, at the additional performance cost of AES256, RMAC can increase security bounds with the "mode 2" construction. The increased bounds also apply to the general construction, but only under the ideal cipher model of proof.

The crux of the matter is the practical value of RMAC/EMAC's additional security assurance. The security bounds of XCBC appear to be adequate for most current applications, especially given a moderate limit on the size of messages, and the full RMAC bounds arguably far exceed the needs of most applications. (For example, as pointed out in [Rog], block cipher encryption modes would begin to leak information far earlier than RMAC.) However, quantities of data that challenge the XCBC security bounds are already conceivable today, and applications with such data may become more prevalent, so in the long term, NIST believes that EMAC is a more prudent choice for general applications.

The draft Recommendation can be refocused to EMAC by setting the length of R to zero in the two general purpose parameter sets: parameter set II for TDES and parameter set III for AES. For applications that desire additional security assurance, RMAC can be retained as an option in parameter sets IV or V, possibly with the "mode 2" construction.

NIST invites further public comments on the specification of a block cipher authentication mode at encrytionmodes@nist.gov.

References

[PR] E. Petrank and C. Rackoff *CBC_MAC for real-time data sources*. Journal of Cryptology, Vol 13 Number 3, Summer 2000.

Public comments, available through http://csrc.nist.gov/encryption/modes/ :

[Bl] John Black, *Comments on the RMAC algorithm.*
[JJV1] É. Jaulmes, A. Joux, and F. Valette, *A remark on the security of RMAC in the related-key security model*.
[JJV2] É. Jaulmes, A. Joux, and F. Valette, *Comments on the security of RMAC as proposed in NIST Draft 800-38B*.
[Kn] L. R. Knudsen, *Analysis of RMAC*.
[Koh] T. Kohno, *Related-Key and Key-Collision Attacks Against RMAC*.
[Ll] J. Lloyd, *An Analysis of RMAC*.
[Rog] P. Rogaway, *Comments on NIST's RMAC Proposal*.
[Wag] David Wagner, *Comments on RMAC*.